



FORENSICS IMAGE BACKGROUND MATCHING
USING SCALE INVARIANT FEATURE TRANSFORM (SIFT)
AND SPEEDED UP ROBUST FEATURES (SURF)

THESIS

Paul N. Fogg II,

AFIT/GCE/ENG/08-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GCE/ENG/08-02

FORENSICS IMAGE BACKGROUND MATCHING
USING SCALE INVARIANT FEATURE TRANSFORM (SIFT)
AND SPEEDED UP ROBUST FEATURES (SURF)

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Computer Engineering

Paul N. Fogg II, B.S.

December 2007

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

FORENSICS IMAGE BACKGROUND MATCHING
USING SCALE INVARIANT FEATURE TRANSFORM (SIFT)
AND SPEEDED UP ROBUST FEATURES (SURF)

Paul N. Fogg II, B.S.

Approved:

/signed/

14 Dec 2007

Dr. Gilbert L. Peterson (Thesis Advisor)

date

/signed/

14 Dec 2007

Maj Michael Veth (Member)

date

/signed/

14 Dec 2007

Dr. Rusty O. Baldwin (Member)

date

Abstract

In criminal investigations, it is not uncommon for investigators to obtain a photograph or image that shows a crime being committed. Additionally, thousands of pictures may exist of a location, taken from the same or varying viewpoints. Some of these images may even include a criminal suspect or witness. One mechanism to identify criminals and witnesses is to group the images found on computers, cell phones, cameras, and other electronic devices into sets representing the same location. One or more images in the group may then prove the suspect was at the crime scene before, during, and/or after a crime. This research extends three image feature generation techniques, the Scale Invariant Feature Transform (SIFT), the Speeded Up Robust Features (SURF), and the Shi-Tomasi algorithm, to group images based on location. The image matching identifies keypoints in images with changes in the contents, viewpoint, and individuals present at each location. After calculating keypoints for each image, the algorithm stores the strongest features for each image are stored to minimize the space and matching requirements. A comparison of the results from the three different feature-generation algorithms shows the SIFT algorithm with 81.21% match accuracy and the SURF algorithm with 80.75% match accuracy for the same set of image matches. The Shi-Tomasi algorithm is ineffective for this problem domain.

Acknowledgements

I would like to thank my wife and father-in-law for the time they spent reviewing this thesis for errors. I would also like to thank my computer lab mates for assisting me in the creation of some of the analyst tools I needed. I would like to express my gratitude to the authors of the algorithm implementations for making the programs available. Finally I would like to acknowledge the great effort my thesis advisor Dr. Peterson and committee members put forth to assist me in getting this thesis completed.

Paul N. Fogg II

Table of Contents

	Page
Abstract	iv
Acknowledgements	v
List of Figures	viii
List of Tables	x
List of Symbols	xi
List of Abbreviations	xii
 I. Introduction	 1
1.1 Research Goal	2
1.2 Sponsor	3
1.3 Assumptions	3
1.4 Organization	3
 II. Background	 5
2.1 Image Matching and Image Registration	5
2.2 Algorithms Used in This Research	8
2.2.1 Scale Invariant Feature Transform (SIFT) Algo- rithm	8
2.2.2 Speeded Up Robust Features (SURF) Algorithm	16
2.2.3 Shi-Tomasi Algorithm	17
2.2.4 Alternative Image Matching Algorithm	17
2.3 Alternative Reduction Method	18
2.4 Summary	19
 III. Keypoint Reduction, Matching and Keypoint Match Comparison	 20
3.1 Keypoint Reduction	20
3.2 Matching Using SIFT	22
3.3 Matching Using SURF	22
3.4 Keypoint Match Comparison	24
3.4.1 SIFT Keypoint Match Comparison	25
3.4.2 SURF Keypoint Match Comparison	27
3.5 Matching Using Shi-Tomasi Algorithm	28

	Page
IV. Results	29
4.1 Image Data Base	30
4.1.1 Testing Keypoints	33
4.2 SIFT Algorithm	35
4.3 SURF Algorithm	37
4.4 Human Accuracy	40
4.5 Shi-Tomasi Algorithm	42
4.6 Results Conclusion	42
V. Conclusion	43
5.1 Future Work	43
Bibliography	45

List of Figures

Figure		Page
2.1.	Dashed line shows the percent of correctly matched keypoints as a function of database size. Solid line shows the percent of keypoints assigned to the correct location, scale, and orientation [11].	10
2.2.	Difference of Gaussians computation. [11]	12
2.3.	Maximum Minimum Difference of Gaussians computation. [11]	12
2.4.	Keypoint Descriptor Generation. [11]	15
3.1.	Example output of the SIFT matching algorithm.	23
3.2.	Example output of the SURF matching algorithm.	24
3.3.	An example matching image showing good candidate for keypoint match quality check.	26
4.1.	Unreduced SIFT Accuracy.	31
4.2.	Unreduced SURF Accuracy.	32
4.3.	Example Keypoint Distribution. Axis are the x, y coordinate of pixels	34
4.4.	Two feature distributions of 102 keypoints with different weighting on distance. Axis are the x, y coordinate of pixels	34
4.5.	SIFT Image Showing The Reduced Keypoint Matches With Occlusion.	36
4.6.	SIFT Error on Reduced Keypoints Showing the False Positives and False Negatives.	36
4.7.	Reduced Keypoint SIFT Accuracy with and without Match Comparison Test.	37
4.8.	Match Comparison Test Showing Match Lines, Intersect Points (Circles), Average Intersect Point (X), And The Standard Deviation Distance (Dotted Line).	38
4.9.	SURF Image Showing Reduced Keypoint Matches With Occlusion.	39

Figure		Page
4.10.	SURF Error on Reduced Keypoints Showing the False Positives and False Negatives.	39
4.11.	SURF Accuracy on Reduced Keypoints.	40
4.12.	SURF Accuracy with $\epsilon = 0.4$	41
4.13.	SURF Accuracy with $\epsilon = 0.3$	41

List of Tables

Table		Page
3.1.	Weightings for Mahalanobis distance and constant scale	21
4.1.	Image groups and number of images in each group	30
4.2.	Image groups and number of compares in each group	30
4.3.	Disk space needed to accommodate the reduced and unreduced algorithm files.	32
4.4.	Reduced and unreduced algorithm match execution time for SIFT.	33
4.5.	Reduced and unreduced algorithm match execution time for SURF.	33

List of Symbols

Symbol		Page
ϵ	This symbol represents the threshold slope used in the SURF Quality Check	28
η	This symbol represents the minimum threshold for indicating a match between two images	31

List of Abbreviations

Abbreviation		Page
SIFT	Scale Invariant Feature Transform	iv
SURF	Speeded Up Robust Features	iv
DCCI	Defense Cyber Crime Institute	3
DOG	Difference of Gaussians	11
PCA	Principal Components Analysis	17
KLT	Kanade-Lucas-Tomasi Feature Tracker	28
POV	Point of View	29

FORENSICS IMAGE BACKGROUND MATCHING
USING SCALE INVARIANT FEATURE TRANSFORM (SIFT)
AND SPEEDED UP ROBUST FEATURES (SURF)

I. Introduction

The use of electronic matching for forensics is common for fingerprints [6], shoe imprints [2], and face recognition [22]. Like these approaches, the Scale Invariant Feature Transform (SIFT) [10], Speeded Up Robust Features (SURF) [4], and the Shi-Tomasi [16] image feature generation algorithms can automate the process of image matching. By matching and grouping multiple images of the same location, even from different viewpoints, it is possible to link suspected criminals and/or their victims to a specific location given their image in different photos. If the specific location is known to be a crime scene, the suspected criminal can be placed at the scene of a crime. If an image shows a victim, such as a child pornography victim, against an unidentified background, matching and grouping of multiple images may establish where the victim was, at the time the image was made. This research contributes to the creation of a powerful tool for law enforcement to combat the heinous crime of child pornography. An important reason to automate this process is to minimize the time a person must spend manually viewing these disturbing images. Humans are subject to fatigue and are prone to error, but computers running an algorithm can run 24 hours a day.

One of the difficulties in automating the processes is occlusion, where people or objects block parts of the image background. Some of the other difficulties include image quality, noise in the image, resolution differences, and large changes in translation and rotation of the camera. Rotation of an object turns or spins an object on its center axis. The rotation of the camera can be thought of as its pitch, yaw, and roll angles. Translation of an object is the moving of the camera in space to

a new position without rotation or reflection. Occlusion complicates identifying the background location. Scale, illumination, and affine transforms of the objects in the scene also complicate matters. This research uses existing algorithms for scale and affine transform issues and to perform matching in the presence of large occlusions.

The process for performing image matching has several components. The first is the generation of keypoints for each algorithm. The second component, needed due to the large number of keypoints produced by SIFT and SURF, is keypoint reduction, which reduces storage requirements and speeds the matching process. The third component is match comparison which removes poor quality keypoint matches between two images. The final component groups images taken at the same location.

When a suspect is discovered and their residence and/or business is search, any computers are seized and brought to the forensics lab and disassembled. The hard drives are acquired and images are extracted from each hard drive and placed in an image data base. Images are converted to SIFT and/or SURF keypoint files. Keypoints files are reduced and saved into a keypoint data base. The match process is run on all the reduced keypoint files. The match files are analyzed using the match comparison algorithms. Image matches are returned for each image based on the number of keypoint matches between two images. The returned images are inspected by a human to determine if a link can be established between a victim and criminal.

1.1 Research Goal

The goal of this research is to advance the knowledge and application of tools in background image matching in the domain of child pornography. This research pays special attention to size reduction of the data bases and the speed of the image matching.

The research focuses on three algorithms that produce interest points (keypoints) SIFT [10], SURF [4], and Shi-Tomasi [16]. One of the hallmarks of the SIFT algorithm is it produces a large number of the keypoints. Each keypoint is a 128 element feature vector, a scale, and an orientation [10]. SURF also produces a large

number of keypoints but has a smaller 64 element descriptor vector which contains entries of the second moment matrix and the sign of the descriptor Laplacian [4] [3]. This research uses 128 element descriptor version of SURF. The Shi-Tomasi algorithm generates a specified number of keypoints with a particular quality value [5].

1.2 Sponsor

This thesis supports the Defense Cyber Crime Institute. The Defense Cyber Crime Institute (DCCI) provides legally and scientifically accepted standards, techniques, methodologies, research, tools and technologies on computer forensics and related technologies to meet the current and future needs of the DoD counterintelligence, intelligence, information assurance, information operations and law enforcement communities.

1.3 Assumptions

This research has some assumptions and limitations. All tests were conducted on a single data base of images that did not contain any actual child pornography images. The usefulness of the research may be limited by the typical backgrounds found within actual child pornography image data bases. Most of the images were of a single resolution. Of the 125 images, 119 are 1600X1200 pixels and 6 are 640X480 pixels. All the images were taken using the same camera with the same jpeg quality settings. A consequence of this is images from different sources and at different qualities may not match as well. The number of keypoints selected for the keypoint reduction was 102. This number of keypoints was not optimized and was determined by trial and error.

1.4 Organization

Chapter II provides background information on each of the feature generation algorithms (SIFT, SURF, and Shi-Tomasi) and image matching techniques. Chapter 3 presents, for each algorithm, keypoints reduction method and the technique used to

identify a location match given the variation in viewpoint, contents, and obstructions. This is followed by a description of the experimental system and results, showing both the best performing algorithm and the best settings for each algorithm. The final Chapter presents conclusions and a discussion of future work.

II. Background

This chapter reviews previous work on image and/or object recognition. Information on the three algorithms used in this thesis is presented. A possible alternative algorithm to the three listed is shown and an alternative keypoint reduction method is presented.

2.1 *Image Matching and Image Registration*

A topic strongly related to image matching is image registration. A 2003 survey [23] provides a good summary of image registration techniques. Image registration is a method where two or more images are taken from the same location. These images can vary in the point of view, can be taken at specific time intervals, and also be taken with different sensors technologies [23]. The images are transformed to the same coordinate system or as the authors state “geometrically aligns two images [23].” Four basic procedures are needed to perform image registration: “feature detection, feature matching, mapping function design, and image transformation and resampling [23].” Within the four basic procedures two area-based and feature-based methods are presented and discussed [23].

Feature detection collects points based on their distinctiveness within the images called control points [23], keypoints [10], and interest points [4]. Area-based methods do not use feature detection. Feature-based methods use three mechanisms for control point generation including Region features, Line features, and Point features. Region features are usually detected by segmentation and other techniques; namely virtual circles [1], Harris corner detector [14] and, Maximally Stable Extremal Regions [12] [23]. Region features are large closed-boundary areas distinctive from the background. Examples include lakes, densely populated areas, and dense wooded areas [23]. Line features are detected by a variety of methods. The Canny detector and a Laplacian of Gaussian based methods are most popular [23]. Line features extract line-like elements within an image such as roads, coastlines, exposed pipelines and, other elongated elements in an image. Point features have the greatest number of detection techniques

including wavelets of various types and, corner detectors. Wavelets methods detect inflection points of curves or local extrema [23].

The second basic procedure of image registration is feature matching. Feature matching is performed on image intensity values using close neighborhoods, spatial distribution, or symbolic descriptions [23]. Area-based methods combines feature detection and feature matching and also performs matching without the identification of distinctive objects. Area-based methods use correlation-like methods, Fourier methods and, Mutual information methods. The correlation-like methods such as cross-correlation, use the image intensities directly without a structural analysis but are sensitive to image noise [23]. The Fourier methods are fast and insensitive to correlated and frequency dependent noise [23]. The mutual information methods are used in mulitmodal registration, and are especially important in medical imaging [23]. Feature-based methods use the correspondence between the interest features by using spatial relations or descriptors associated with the interest feature. Feature-based methods include methods using spatial relations, invariant descriptors, relaxation methods and, pyramids and wavelets [23]. Methods using spatial relations take advantage of the properties of the distribution of the interest features. Methods using invariant descriptors have important conditions to be met. The descriptors should be invariant, unique, stable, and independent [23]. Both the SIFT and SURF algorithms use invariant descriptors method [4] [10]. The relaxation method is based on a solution to the consistent labeling problem which labels each feature from one image with the label of a feature from the image to be matched. Then the consistent labeling problem iterates through the possible solutions until a stable one is found [23]. Pyramids and wavelets in feature matching reduces computational time. This method is designed to hierarchically traverse the resolution from coarse to fine [23].

The third basic procedure of image registration is transform model estimation. Transformation types include global mapping models, local mapping models, mapping by radial basis functions, and elastic registration [23]. Global mapping, as the name implies, performs the mapping function on the image as a whole. Global mapping

models include similarity transform, affine transform, and perspective projection. The local mapping models take care of images with local deformations. The local methods include, weighted least square, weighted mean, piecewise linear mapping, piecewise cubic mapping and Akima's quintic approach [23]. Radial basis functions are a subset of global mapping function that have the ability to deal with local deformations. Examples of radial basis functions are "multiquadrics, reciprocal multiquadrics, Gaussians, Wendland's functions, and thin-plate splines [23]." In the elastic registration method, images are seen as thin flexible sheets that can be bent, stretched, and molded into the reference image [23].

The fourth and final step in image registration procedure is image resampling and transformation [23]. This technique alters the image to be matched so the image can be overlayed with the reference image. The interpolation is normally implemented with convolution using several kernel functions such as nearest neighbor function, bilinear function, bicubic functions, quadratic splines, and Gaussians to name a few [23].

In contrast, interest or salient points must be distinctive and invariant to the expected deformations considered. Many interesting points in images are not corners but are contained in smooth areas [15]. The research introduces a wavelet-based interest point detector. Using this interest point detector with the wavelet functions Haar, Daubechies4, Harris, and PreciseHarris the information content or entropy is computed giving 6.0653, 6.1956, 5.4337, and 5.6975 respectively [15].

A similar problem domain to this research which is forensic based and image matching is presented in [13]. This paper focuses on the improvement of the accuracy of the ranking of cartridge cases when searching for breech face marks and firing pins in a data base of forensic images [13]. A preprocessing step using the Kanade-Lucas-Tomasi equation to provide a fast pre-selection method is used for accuracy improvement [13].

2.2 Algorithms Used in This Research

2.2.1 Scale Invariant Feature Transform (SIFT) Algorithm. The SIFT algorithm was developed by David G. Lowe [10] and performs image recognition by calculating a local image feature vector. The feature vector is invariant to image scaling, translation, rotation, and partially invariant to changes in illumination and affine transformations [10]. The features were inspired by the “neurons in inferior temporal cortex that are used for object recognition in primate vision [10].” The calculation of the features occurs in a multiphased filtering process that discovers interest points in scale space. Keypoints are generated which account for the local geometric deformations by characterizing blurred image gradients in numerous orientation planes and at various scales [10]. The keypoints are used as input to a nearest-neighbor indexing procedure that ascertains potential object matches between two images. “A low-residual least squares solution for the unknown model parameters [10]” is computed for each match which ensures matches are high quality. The algorithm produces a large number of keypoints that allow robust object recognition in cluttered partially-occluded images [10]. Prior to the SIFT algorithm, local feature generation was not invariant to scale and had a greater sensitivity to affine transformations, and rotation. The SIFT algorithm also emulates biological systems in feature generation, which renders features partially invariant to local variations by blurring image gradient locations. An example of local variations are affine transformations which are based on a model of the complex behavior of cells within the cerebral cortex of mammalian vision [10]. Extraction of keypoints is performed using the following four steps:

1. Scale-space extrema detection: This stage successively blurs the images convolved with an increasing variance Gaussian kernel. The Difference-of-Gaussian function is computed from the resulting octave of blurred images. From the difference-of-Gaussian the potential interest points are identified using a corner detection threshold [11].

2. Keypoint localization: At each potential keypoint, the model is refined to determine location and scale. The final keypoints are selected based on measures of their stability. The points that do not meet the stability requirements are pruned [11]. This step begins by using a quadratic least square fit to refine the location of the detected extrema [9].
3. Orientation assignment: One or more orientations may be assigned to each keypoint based on local image gradient directions. With the orientation assigned, all future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature. This provides invariance to these transformations [11].
4. Keypoint descriptor: Local image gradients are measured at the selected scale in a 16x16 pixel patch around each keypoint. This information is transformed into vectors 128 elements long that allow significant levels of local shape distortion and change in illumination [11].

Matching reliability is an issue with a large database of keypoints from 112 images [11]. To underscore this point, Figure 2.1 shows a marked decrease in matching performance with nearest descriptor in database repeatability percentage. This decrease is with 100,000 keypoints from 112 images [11]. This is an important factor in the problem domain of matching a large number of images to an even larger number of images in a database. For a child pornography case, there are potentially tens of thousands of images collected with an average of 3000 keypoints for each image, resulting in more than 30,000,000 keypoints. These are the major reasons for finding an effective method of reducing the number of keypoints used in the matching of images and improving the matching technique.

The SIFT matching process uses the Euclidean distance between the descriptor vectors of keypoints to determine if there is a possible match. The use of a nearest neighbor approximation called Best-Bin-First avoids an exhaustive search of the keypoint data base [11]. Search time is reduced by ending the search when the first 200

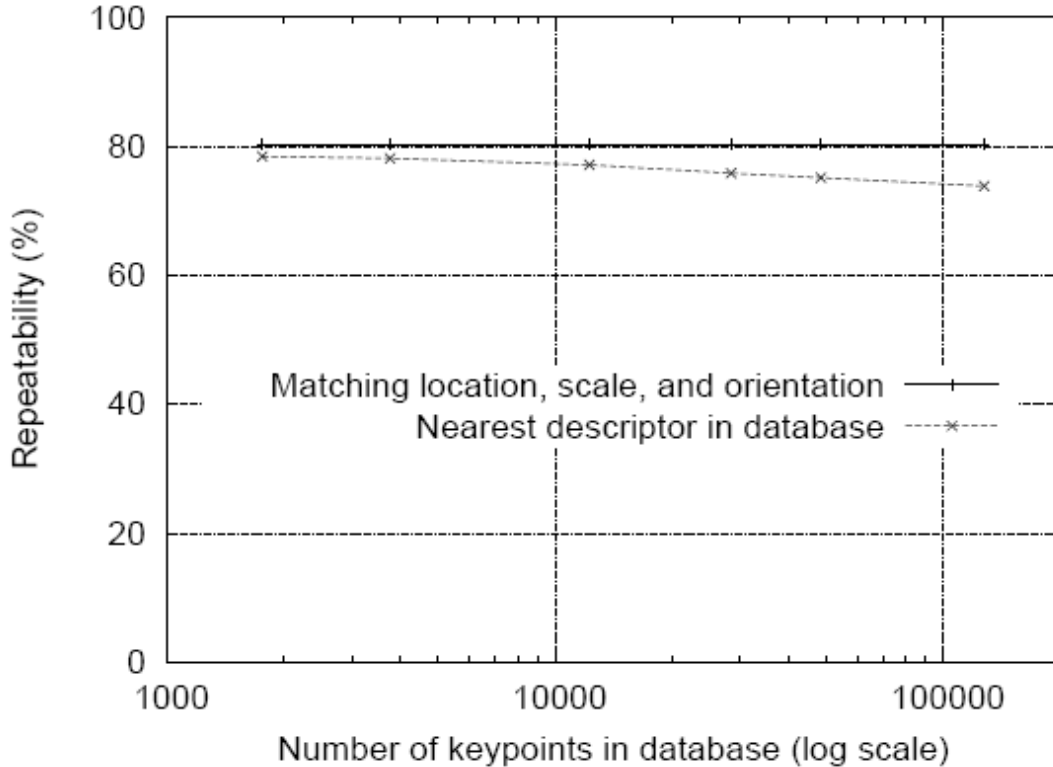


Figure 2.1: Dashed line shows the percent of correctly matched keypoints as a function of database size. Solid line shows the percent of keypoints assigned to the correct location, scale, and orientation [11].

nearest neighbor candidates are checked [11]. The Best-Bin-First “algorithm works particularly well for this problem is that we only consider matches in which the nearest neighbor is less than 0.8 times the distance to the second-nearest neighbor [11]” so an exact solution is not required. With object recognition, as few as 3 feature matches are needed using the Hough transform to find clusters of features. An affine solution is used to account for affine transformations and is solved using the least squares solution [11].

The following subsections go into more detail on the steps in the generation of SIFT keypoints.

2.2.1.1 *Scale-space Extrema Detection.* The scale-space kernel [11] is a Gaussian function and is

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.1)$$

where the x and y are related to the blur radius r by $r^2 = x^2 + y^2$ and σ is the scale. The scale space function is

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.2)$$

where $I(x, y)$ is the gray scale image, and $*$ is the convolution operation in x and y [11]. A factor k is multiplied by σ where $k = 2^{1/s}$ and s is an algorithm parameter such that $s + 3$ is the number of images in an octave. For each octave the images dimensions are halved which produces a pyramid of Gaussian filtered images [9]. With the Gaussian filtered images computed, the Difference of Gaussians (DOG) is computed or

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \quad (2.3)$$

An example DOG can be seen in Figure 2.2. The next part of the scale-space extrema detection step is the local extrema detection. The pixels that pass this test are called keypoints and are found by identifying all of the pixels that correspond to extrema of $D(x, y, \sigma)$ [9] where the maximum and minimum locations are. “In order to detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below (see Figure 2.3). It is selected only if it is larger than all of these neighbors or smaller than all of them. The cost of this check is reasonably low due to the fact that most sample points will be eliminated following the first few checks [11].” Figure 2.3 shows maxima and minima of the DOG images detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles) [11].

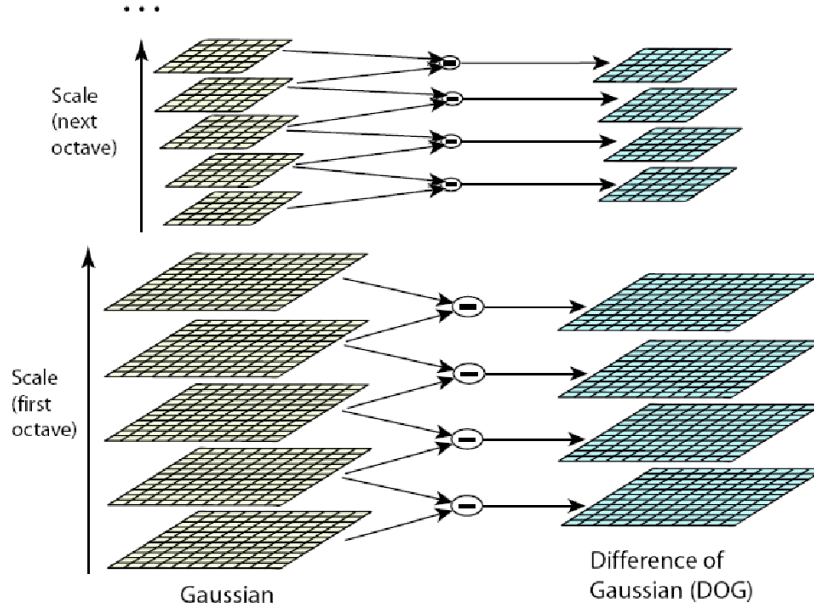


Figure 2.2: Difference of Gaussians computation. [11]

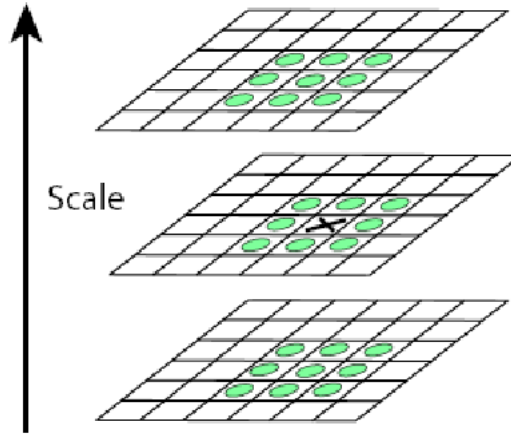


Figure 2.3: Maximum Minimum Difference of Gaussians computation. [11]

2.2.1.2 Keypoint Localization. This step uses a quadratic least square fit to refine the location of the detected extrema [9]. The first substep extracts the

2nd order Taylor expansion of D at (x, y, σ) . This expansion yields

$$D(\Delta x, \Delta y, \Delta \sigma) = \frac{1}{2} [\Delta x \ \Delta y \ \Delta \sigma] \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial x \partial \sigma} \\ \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y \partial \sigma} \\ \frac{\partial^2 D}{\partial x \partial \sigma} & \frac{\partial^2 D}{\partial y \partial \sigma} & \frac{\partial^2 D}{\partial \sigma^2} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \sigma \end{bmatrix}. \quad (2.4)$$

Using vector notation for (2.4) yields

$$D(\Delta \vec{x}) = D(\vec{x}) + \left(\frac{\partial D}{\partial \vec{x}} \right)^T \Delta \vec{x} + \frac{1}{2} (\Delta \vec{x})^T \frac{\partial^2 D}{\partial \vec{x}^2} (\Delta \vec{x}) \quad (2.5)$$

where \vec{x} is

$$\vec{x} = \begin{bmatrix} x \\ y \\ \sigma \end{bmatrix} \quad (2.6)$$

and

$$\Delta \vec{x} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \sigma \end{bmatrix}. \quad (2.7)$$

The derivatives of (2.5) with respect to $\Delta \vec{x}$ are solved for $\frac{\partial D}{\partial \Delta \vec{x}} = 0$ which results in

$$(\Delta \vec{x}) = - \left(\frac{\partial^2 D}{\partial \vec{x}^2} \right)^{-1} \left(\frac{\partial D}{\partial \vec{x}} \right) = \hat{x}. \quad (2.8)$$

A location refinement is performed if $\hat{x} > 0.5$ in any dimension. When a location refinement is required the point is moved to $(x + \Delta x, y + \Delta y, \sigma + \Delta \sigma)$ [9]. The weak keypoints are removed if $|D(\hat{x})| < 0.03$ for pixel values in the $[0,1]$ range [11] or by eliminating edge responses by computing the 2x2 Hessian matrix

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.9)$$

and then the trace

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (2.10)$$

and the determinate

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta. \quad (2.11)$$

In (2.10) and (2.11), α is the eigenvalue with the largest magnitude and β is the eigenvalue with the smallest magnitude. letting $\alpha = r\beta$, then

$$\frac{Tr(H)^2}{Det(H)} = \frac{(r+1)^2}{r} \quad (2.12)$$

If the ratio of principal curvatures are less then the RHS of

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad (2.13)$$

For $r = 3$ [11], the keypoint is also pruned.

2.2.1.3 Orientation Assignment. In this step, an orientation is assigned to each keypoint. For each image sample, $L(x, y)$ at the closest scale, the gradient magnitude and the orientation are respectively

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}, \quad (2.14)$$

and

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}. \quad (2.15)$$

A histogram is built from the gradient orientations of the neighbors of a keypoint. The histogram has 36 slots representing the 360 degrees possible. Each point sampled from around the keypoint is weighted by its gradient magnitude and with a Gaussian-weighted circular window with σ equal to 1.5 times the scale of the keypoint [11]. The highest peak in the histogram, and all other peaks within 80% of the highest peak,

are set as the orientation of the keypoint. To improve accuracy, a parabola is fit to the three histogram values that are closest to each peak [11].

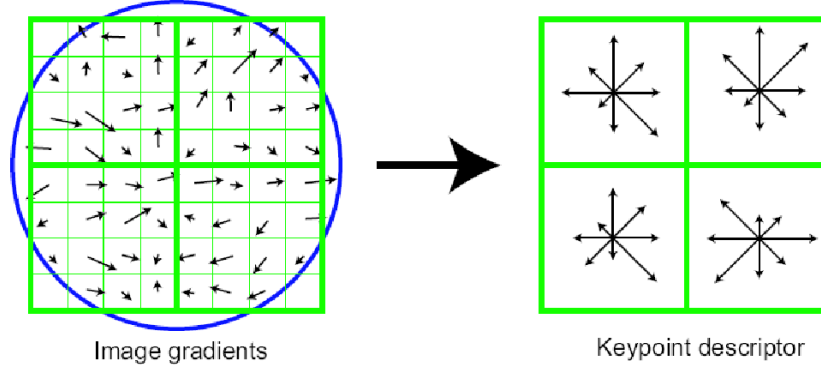


Figure 2.4: Keypoint Descriptor Generation. [11]

2.2.1.4 Keypoint Descriptor. A keypoint descriptor is created by computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left of Figure 2.4. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are accumulated into orientation histograms over 4x4 subregions, as shown on the right of Figure 2.4. The length of each arrow corresponds to the sum of the gradient magnitudes near that direction within the region [11]. Figure 2.4 is a 2x2 descriptor array computed from an 8x8 set of samples, whereas the procedure herein uses a 4x4 descriptor computed from a 16x16 sample array. Using this method, a descriptor of $4 \times 4 \times 8 = 128$ elements is obtained 4x4 descriptors and 8 bins. Descriptors are generated using a 16x16 image patch from $I(x, y)$ and (2.1) or compactly written as $I * G_{\sigma_i}$ where σ_i is the scale of the keypoint centered at (x_i, y_i) . Next, the gradient orientation relative to keypoint's orientation is computed followed by the orientation histogram of each 4x4 pixel block. Due to the Gaussian weighted window, pixels closer to the center of the 16x16 patch contribute more to the orientation histograms. Keypoint descriptors are generated by:

1. Computing gradients,

2. Computing relative gradient orientations,
3. Defining an accumulator variable for each of the 8 orientations in each of the 16 histograms (128 total), and finally
4. For each pixel, calculate the pixels contribution to each accumulator variable [9].

Some post processing is needed to obtain invariance to linear lighting variations and reduce the weight to very large gradient magnitudes. Post processing steps are normalized to f , where f is the feature vector. All elements in the feature vector are clamped to 0.2. Finally, f is re-normalized [9].

2.2.2 Speeded Up Robust Features (SURF) Algorithm. The SURF algorithm improves upon the currently used scale and rotation invariant interest point or keypoint detector and descriptor [4]. Bay, et al. used several new techniques that increased the speed of the interest point detector, the descriptor generation, and the matching. Hessian-based detectors such as SURF are stable, repeatable, and fire less on elongated, ill-localized feature structures [4]. The use of approximations, as in the DOG in SIFT [11], gives a speed advantage with a relatively low negative impact on accuracy [4]. SURF uses a Fast-Hessian Detector to generate interest points or keypoints. That is, the determinant of the Hessian provides a metric to select the location and the scale points. For the blurring step of the calculations, SURF uses an approximate second order Gaussian derivative using box filters which increases the algorithms performance. The runtime in experiments of the keypoint detector and descriptor generation are 354ms, 391ms, and 1036ms for SURF, SURF-128, and SIFT, respectively [4]. The average recognition rates or accuracy of detecting a repeat location for each algorithm is SURF 82.6%, SURF-128 85.7%, and SIFT 78.1% [4]. The SURF descriptor has similar properties to the SIFT descriptor but it is less complex than the SIFT descriptor, only requiring two steps to construct. The first step in building a descriptor is finding a reproducible orientation based on the information around the region near the interest point. Next, a square region is generated and aligned to the selected orientation, and the SURF descriptor is generated [4].

2.2.3 Shi-Tomasi Algorithm. The Shi-Tomasi algorithm [16] selects features that are suitable for tracking between image frames. The method used in the generation of keypoints or interest points is simpler than SIFT or SURF. Consequently, the method is not as robust to large displacements because it was designed for the relatively small changes found in the sequential frames of a video. The calculation of the Shi-Tomasi keypoints start by breaking the image into 7X7 patches and computing the second order partial derivatives from $g(x, y)$, where $g(x, y)$ is the intensity function. From the second order computation, the Hessian matrix is

$$Z = \begin{bmatrix} g_{xx} & g_{xy} \\ g_{xy} & g_{yy} \end{bmatrix} \quad (2.16)$$

Given two eigenvalues of Z are λ_1 and λ_2 , the window is accepted as an interest point if $\lambda < \min(\lambda_1, \lambda_2)$ where λ is a threshold between 1 and x , where x is a number that depends on the desired number of features. As x increases, the number of features retained decreases. [16]. The larger the $\min(\lambda_1, \lambda_2)$ is, the stronger the feature [16].

2.2.4 Alternative Image Matching Algorithm. There are other algorithms available for matching images. One such algorithm, also derived from SIFT, is Principal Components Analysis (PCA)-SIFT [17]. The primary difference is instead of using smoothed weighted histograms to generate the 128 element feature vector, PCA is applied to the normalized gradient patch. This reduces the size of the feature vector to a user specified size. The default feature vector size is 20 [17]. Experiments show SIFT runs slightly faster when generating its keypoints and descriptors, that is, 1.59 sec vs. 1.64 sec [17]. The experiments also show a large performance advantage for PCA-SIFT, 0.58 sec vs. 2.20 sec for the matching portion of the algorithms [17]. The accuracy of PCA-SIFT with a feature vector size of 20 is better than SIFT's accuracy, 68% PCA-SIFT vs. 43% for SIFT [17].

An algorithm has been specifically designed to be invariant to non-affine image deformations [7]. Geodesic sampling is used on an intensity image embedded as a sur-

face in 3D space. A third coordinate is defined as the proportion of the intensity values and an aspect weight (α) [7]. As α approaches 1 “the geodesic distance is exactly deformation invariant [7].” Once the sample points are acquired the “geodesic-intensity histogram [7]” is built and defined as the local descriptor.

New methods of algorithm modularization, tuning and testing are developed in [21]. An image data base which has known, accurate match information, images with 3D variations which provide more complex images than planar-based testing is generated. The descriptor generation process is split into modules which enable different descriptor combinations. Thus, a composite descriptor generations algorithm can be developed and also modules can be analyzed in finer detail. Learning methods are used to tune the various parameters within each module to provide improved performance [21].

2.3 Alternative Reduction Method

An alternative method to reduce keypoint file size is presented in [20]. The reduction method takes advantage of the characteristics of an indoor environment which is assumed. The most important assumption is the stability of the viewpoint of the images with minimal rotation [20]. Three rotational invariance steps are removed from the SIFT algorithm thus reducing the computational requirement of the algorithm [20]. The three steps are as follows:

- “The calculation and assignment of keypoint orientations [20]”
- “The generation of additional keypoints at locations with multiple dominant orientations [20]”
- “The alignment of the keypoint descriptor to the keypoints orientation [20].”

This method also reduces the number of keypoints generated since more than one keypoint can be produced if there is more than one peak generated during orientation assignment [20]. The reduction method is effective for images taken with very little rotation [20].

2.4 *Summary*

Previous work on image recognition and image registration was examined. Another forensic image matching domain, firearm ballistics information, was presented. Information was provided on the three algorithms, SIFT, SURF, and Shi-Tomasi, used in this thesis. Alternative algorithms to the three listed were shown and an alternative keypoint reduction method was presented.

III. Keypoint Reduction, Matching and Keypoint Match Comparison

This chapter presents the methods for keypoint reduction, matching and keypoint match comparison. Varying methods were required to execute each of these components to improve the overall matching efficiency. The space needed to store the keypoint or interest point files was reduced and the speed of this matching process increased.

3.1 Keypoint Reduction

Since SIFT and SURF generate an average of 3000 keypoints per image reducing the number of keypoints generated is desirable. This reduction will minimize the SIFT or SURF keypoint data base and speed up the matching process. The matching must occur across multiple keypoint data sets and different individuals picture collections, so matching speed is important. However, reducing the keypoints could have a dramatic impact on the matching accuracy. To counter this, a technique to choose stronger keypoints from the thousands of keypoints produced is developed. A distance function ensures a good keypoint spread. Once the keypoints are computed and reduced, space requirement is reduced and computation time is reduced for all future matching runs.

Keypoints are selected using an iterative approach. Specific to the SIFT algorithm the first two points selected are based upon the scale of the detected keypoints. The SURF algorithm selects the first two points based on non zero elements of the second moment matrix. Consequent keypoints are selected for SIFT based on a weighted sum of the scale of the keypoint and of the Mahalanobis distance between it and all of the other chosen points [18] [19]. Keypoints are obtained by evaluating each available point (x_i, y_i) using

$$W_1 D_{mahal}(x_i, y_i) + W_2 \sigma(x_i, y_i) \quad (3.1)$$

to get the highest value where $\sigma(x_i, y_i)$ is the scale, $D_{mahal}(x_i, y_i)$ is the Mahalanobis distance at point (x_i, y_i) , W_1 is the weighting of the Mahalanobis function, and W_2

is the weighting of the scale of the keypoint. This process continues until the desired number of keypoints are selected. The Mahalanobis distance function is a weighted Euclidean distance with the weighting defined from the sample variance-covariance matrix. Experimentation shows that the computational overhead increases as the number of keypoints selected increases, as expected.

The best weight values for the weighting between distance and scale was determined using the weights and scale shown in Table 3.1.

Table 3.1: Weightings for Mahalanobis distance and constant scale

Mahalanobis Distance Weight	Constant Scale Weight
0.5	1
1	1
5	1
10	1
50	1
100	1

The goal is to ensure that selected keypoints are spread uniformly to prevent partial occlusion yet still provide a strong probability of matching. As the weighting on the distance increases, keypoints begin to cluster while equal weights result in a better distribution of keypoints. With a weighting of 0.5, the distribution of keypoints was virtually unchanged from a distance weighting of 0. This was determined subjectively by overlaying the keypoint distributions and observing the levels of spread and clustering. Therefore, the weights for distance and scale are both set to 1. With the SIFT algorithm, the scale and distance were the deciding factor on the quality of the keypoint. The SURF algorithm does not provide a scale component and so the non-zero element of the second moment matrix is used as the scale component. This second moment is calculated over the cardinality of the non-zero elements (Nz) using

$$\log \left(\frac{1}{\sqrt{|Nz|^2}} \right). \quad (3.2)$$

The techniques used for image matching are discussed next using the reduced set of keypoints computed in this section.

3.2 Matching Using SIFT

Image background matching for SIFT is an extension of the SIFT implementation of Rob Hess [8]. The modified program allows more command line arguments to change image inputs, alter parameters, allow the matching of SIFT files using the keypoint descriptor files, and saving matched points to a text file for further analysis. For each match, an image file that shows the matches produced. A batch file cycles through the different SIFT keypoint descriptor files, image files, and names the output text files. The matching algorithm is computed using the methods described in [11] by using the best candidate match of keypoints. The best candidate match is found by calculating the nearest neighbor using a minimum Euclidean distance for the descriptor vector. The second-closest neighbor's distance defines the distance ratio where, with a distance ratio greater than 0.8, 90% of the bad matches are pruned [11]. The Best-Bin-First algorithm approximates the search of the nearest neighbor [11] and the Hough transform identifies clusters of features to increase recognition of small or occluded objects [11]. Figure 3.1 shows the SIFT matching program output. Each line indicates a keypoint match between the images. It can be seen that there are a large number of matches - 427 to be exact.

3.3 Matching Using SURF

The data sets generated are fed through a MATLAB[®] implementation of the matching process implemented by D. Alvaro and J.J. Guerrero [3]. The matching process begins by loading the two images into memory. SURF keypoints are computed for each image and stored. The approximate distances of the descriptors are computed and sorted using the dot products between unit vectors. A match is accepted if its distance, represented by the ratio of vector angles, is within the distance ratio of 0.95 [3]. The MATLAB[®] files were modified to skip the SURF keypoint computation

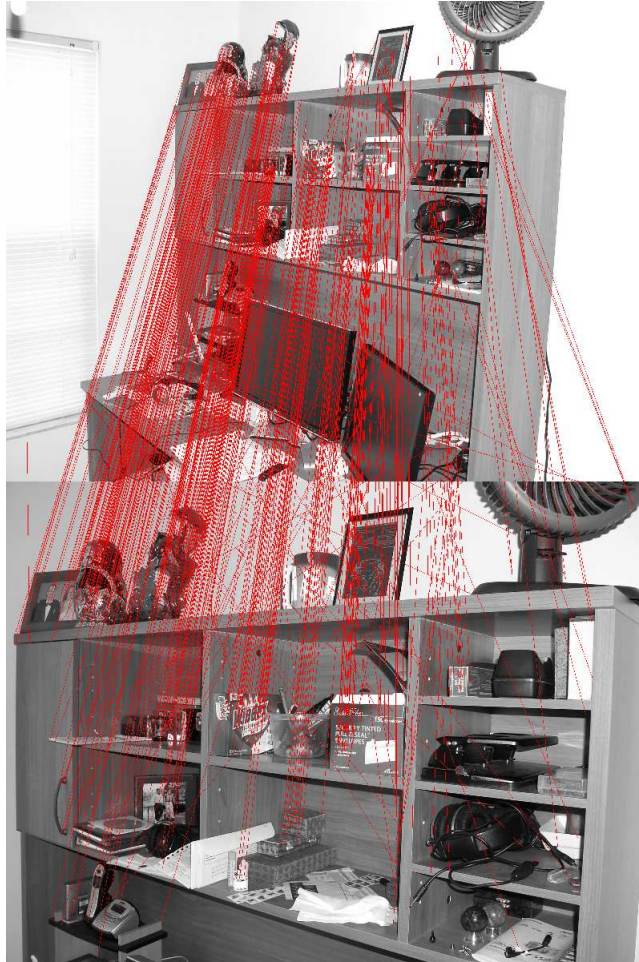


Figure 3.1: Example output of the SIFT matching algorithm.

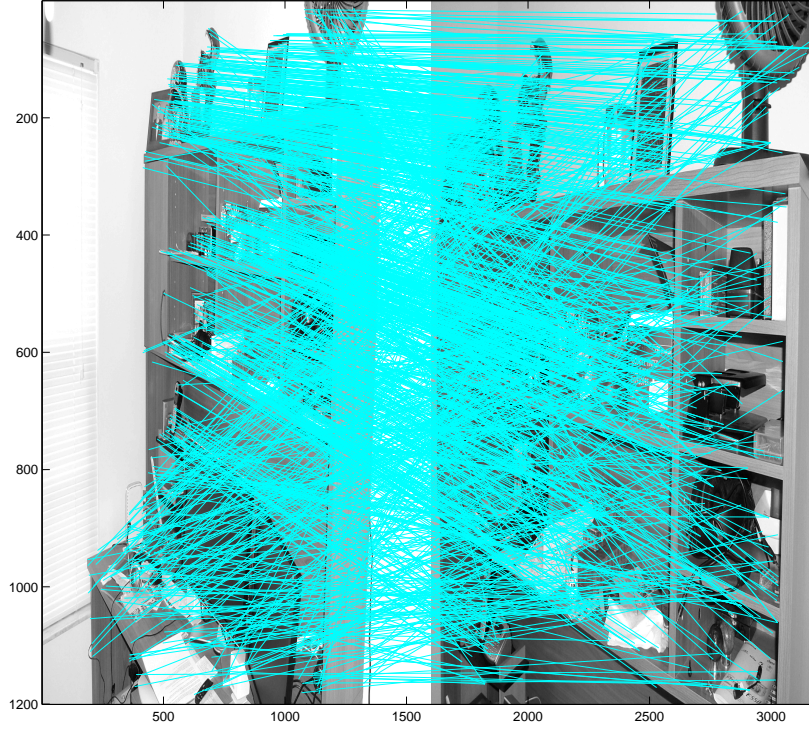


Figure 3.2: Example output of the SURF matching algorithm.

and load the keypoint file instead so the reduced interest point files can be used. Figure 3.2 shows the output of the MATLAB[®] implementation of the SURF algorithm with no keypoint reduction and with a total of 1146 keypoint matches found. Like the SIFT algorithm a complete set of matching was performed with this algorithm.

3.4 *Keypoint Match Comparison*

Additional post processing improves the accuracy of both SIFT and SURF matching algorithms for this problem domain. This post processing uses SIFT and SURF identified points that represent the match lines in the output images. Three strategies are used: in-frame intersection match line removal, intersections outside standard deviation distance, and steep angle removal. These are discussed in Sec-

tions 3.4.1 and 3.4.2. RANdom SAmple Consensus algorithm was considered but due to the computational complexity faster algorithms were developed.

3.4.1 SIFT Keypoint Match Comparison . Two quality checks are performed to identify poor keypoint matches during post processing. Both use the same initial steps. First, all match points are imported from an output file and match points (x_1, y_1) and (x_2, y_2) are converted into lines representing the match lines using the equation for a line.

$$y = mx + b, \quad (3.3)$$

the slope

$$m = \frac{y_2 - y_1}{x_2 - x_1}, \quad (3.4)$$

and the y-intercept

$$b = y_1 - (mx_1). \quad (3.5)$$

The intersections of all the match lines are computed by checking if the lines are parallel. If the slope of the first line m_1 is equal to the slope of the second line m_2 the lines are considered parallel. When the lines are not parallel, x is computed using

$$x = \frac{-(b_1 - b_2)}{m_1 - m_2}. \quad (3.6)$$

Using x from (3.6), possible values of y are determined from

$$y_1 = m_1x + b_1, \quad (3.7)$$

and

$$y_2 = m_2x + b_2. \quad (3.8)$$

The y_1 and y_2 values are compared to ensure a correct intersect point. If y_1 and y_2 are equal, then the point (x, y_1) is added to the list of intersect points. The two methods use the intersection points to find keypoint matches that should be removed. The



Figure 3.3: An example matching image showing good candidate for keypoint match quality check.

first method uses the simple idea that a keypoint match is a bad match if it causes intersect points within the frame of the match image. Figure 3.3 is a good example of a match image that shows where this algorithm works well. There is a keypoint match that traverses the frame diagonally intersecting several other keypoint matches. This routine marks the diagonal match line as bad due to all its intersects being within the frame, and excludes the match from the match count total.

The second SIFT keypoint match quality check has the same initial steps as the previous match check. The difference is after all the intersects are computed, a mean intersect point is computed by computing the average x and average y of all

the intersect points (x, y)

$$\begin{aligned}\bar{x} &= \sum_{i=0}^N \frac{x_i}{N}, \\ \bar{y} &= \sum_{i=0}^N \frac{y_i}{N}.\end{aligned}\tag{3.9}$$

With the average intersect point computed (\bar{x}, \bar{y}) , the average distance from the mean and standard deviation of the intersect points is

$$\begin{aligned}\bar{d} &= \sum_{i=0}^N \frac{\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}}{N} \\ \sigma &= \sqrt{\sum_{i=0}^N \frac{(d_i - \bar{d})^2}{N}}.\end{aligned}\tag{3.10}$$

The distance of the intersect points of each line from the average intersect point (\bar{x}, \bar{y}) is checked against σ and accumulated when the distance is greater than σ . The line is marked as bad and ignored if the accumulated intersect points are 90% or more than the total number of intersect points associated with a line. As with the previous keypoint match quality checking algorithm, Figure 3.3 is a good example of a match image that would benefit from this keypoint match quality checking algorithm.

Both algorithms are run on all 7875 matches from a set of 125 images, resulting in 7875 comparisons with a separate output file for the two algorithms. The results of both algorithms are discussed in Chapter IV. The next section explains an additional algorithm to improve SURF algorithm matching for this domain.

3.4.2 SURF Keypoint Match Comparison . The SURF algorithm performs poorly with both of the keypoint match quality checks developed for the SIFT algorithm. The accuracy of SURF has a marked decrease when using in frame intersection match line removal due to the large number of match lines crossing within the frame. The intersection outside standard deviation distance match comparison shows no

change in accuracy for SURF due to the very large standard deviation distances computed for the match lines. None of the match lines fell outside the standard deviation distance. Chapter IV discusses the findings when using the match pruning method used for SIFT. Due to the previous methods not working, a simple but powerful method was developed to adjust for the properties that SURF matches display. From the SURF image in Figure 3.2 it can be seen that the matches cross each other within the image. To remove many of the more obvious mismatches a conditional check is added to remove a match when the slope computed for the match line, exceeds the threshold

$$\left| \frac{y_2 - y_1}{x_2 - x_1} \right| > \epsilon \quad (3.11)$$

where ϵ represents the threshold slope. Test runs used values of 0.3 and 0.4 for ϵ which speeds the processing due to matches not being added to the total match count and the intersect points not computed from Sections 3.4.1 and 3.4.2. Once this pre-pruning process completes, the procedures described in Section 3.4.1 determine valid matches. Results of the application of this technique are discussed in Chapter IV.

3.5 Matching Using Shi-Tomasi Algorithm

Matching with the Shi-Tomasi algorithm is implemented using the Kanade-Lucas-Tomasi Feature Tracker program or (KLT) with a few modifications [5]. The first step in this feature tracking program imports the two images similar to the cartridge matching in [13]. Next a file containing the feature list of the first image is read into memory which increases the speed of the match because the feature computation only needs to be completed for the second image. The performance does not meet SIFT matching output. A reduced set of 125 matches was performed with one image matched against all other images.

IV. Results

This Chapter discusses in detail the research results. This includes the accuracy and errors of the unreduced keypoints for the SIFT and SURF algorithms. This Chapter also covers the results of the reduced keypoints matching, including the matching results with the additional match comparison post processing. The experiment consists of four primary steps. First, for the test set of images, feature points are collected for each of the three different algorithms: SIFT, SURF, and Shi-Tomasi. The second step reduces keypoints using the method presented in Chapter 3. After reduction, the keypoints are stored in a data file for later matching. After matching, a keypoint comparison is performed on the matched keypoints lines in an effort to prune “*bad*” matches. The Shi-Tomasi algorithm did not produce a set of match lines so the final step is not performed. The experiment uses 125 images with resolutions of 1600X1200 and 640X480 from 6 locations all converted to gray scale. The three matching algorithm implementations require gray scale images due to the implementations using the image intensity function $I(x, y)$.

With 125 images there are $\binom{125}{2} = 7750$ comparisons. All comparisons are computed to verify the accuracy of the techniques. The same camera, a Fuji FinePix E550 was used to acquire all images. One hundred nineteen of the images have a resolution of 1600x1200 pixels. Six of the images have a reduced resolution of 640x480 from the same location while the 125 images cover 6 different locations. The locations are a home office, a guest bedroom office, a stairwell, a living room, a home exterior, and a computer lab. For the purpose of match testing, the images were broken up into 7 groups. The home office is split into 2 groups because the viewpoint of the camera is 180 degrees off. Table 4.1 lists the number of images in each of the 7 groups. All locations have images taken from varying vantage points within that space and with different Points Of View (POV). There were a number of widely different viewpoints for each location. The camera distance from the subject for the indoor images was between 2.75 feet and 11 feet, and the rotation varied approximately ± 15 degrees. The angle of the camera from the subject varied more than ± 50 degrees. The home

Table 4.1: Image groups and number of images in each group

Group	Number of Images Assigned
Home Office 1	38
Home Office 2	14
Computer Lab	27
Outside	32
Guest Room	3
Stairwell	3
Living Room	8

office was the only location that had 2 different resolutions, 1600X1200 and 640X480. The outdoor images have much larger variations. The POV of the images varied 50 feet with ± 10 degrees rotation and over ± 180 degree cardinal direction change.

4.1 Image Data Base

Table 4.1 shows the grouping with the number of images in each group. Table 4.2 shows the grouping again but with the number of unique compares for each group. thus, the total number of correct matches excluding images matching themselves is 1675. Approximately 1300 of the image matches of the total shown in Table 4.1 are

Table 4.2: Image groups and number of compares in each group

Group	Number of Compares
Home Office 1	703
Home Office 2	91
Computer Lab	351
Outside	496
Guest Room	3
Stairwell	3
Living Room	28
Total	1675

outside the capability of both SIFT and SURF algorithms to make a positive match. The matching accuracy of SIFT is approximately 50% with ± 50 degree change in view point [11]. SURF matching accuracy is close to that of SIFT [4]. In the interest

of simulating the real world, an ideal data base was not used. An ideal data base would be a set of images that would have an accuracy of 100% for both the SIFT and the SURF algorithms. Testing the unreduced algorithms, both SIFT and SURF, shows positive interesting results. The maximum accuracy for the SIFT algorithm is 81.55% at threshold (η) values of 139 and 140 with a test range of $\eta = 1$ through 300 where η is the minimum threshold for indicating a match between two images. Figure 4.1 shows the accuracy of the SIFT algorithm as the threshold (η) value is adjusted. The maximum accuracy for the SURF algorithm was 78.26% at threshold (η) values

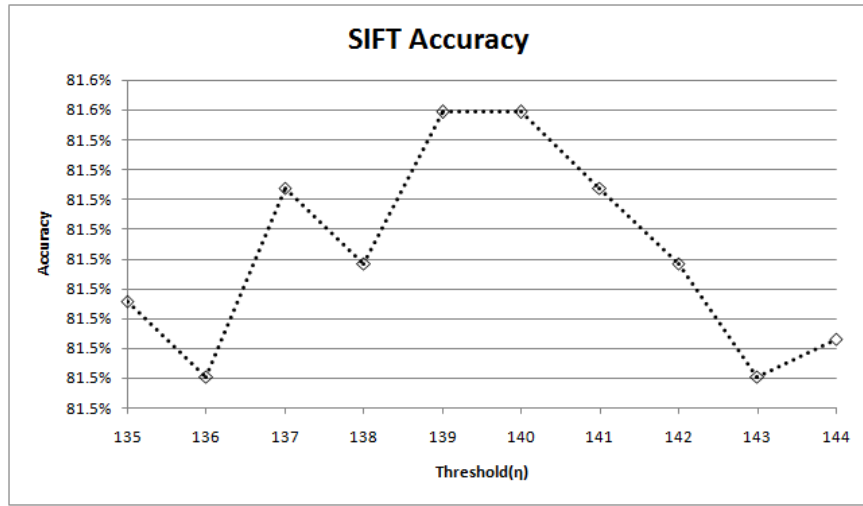


Figure 4.1: Unreduced SIFT Accuracy.

of 1351 through 1363 and 1365 where the test range of η is from 1160 to 1400. Figure 4.2 gives the accuracy of the SURF algorithm with respect to the threshold (η) values. One of the major reasons for reducing the number of the keypoints saved for each image is to conserve storage space. Table 4.3 shows the breakdown of the storage space needed before and after keypoint reduction for both the SIFT algorithm and the SURF algorithm. The size is from the 125 SIFT keypoint files and 125 SURF key files generated from the 125 images. The 97.5% and 94.4% reduction for SIFT and SURF respectively is significant storage space savings. Reduction also reduces the time to compute matches. Table 4.4 lists the time required to run a complete matching experiment for the SIFT algorithm. The time needed to run the matching

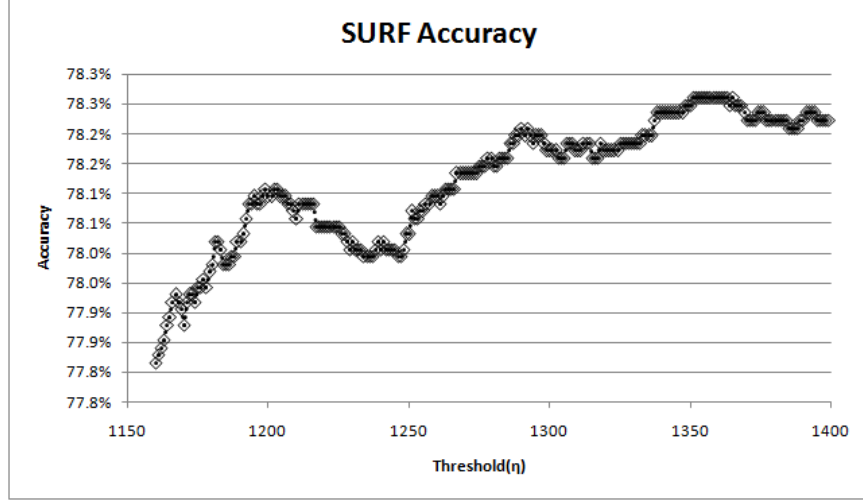


Figure 4.2: Unreduced SURF Accuracy.

Table 4.3: Disk space needed to accommodate the reduced and unreduced algorithm files.

Algorithm	Size On Disk	Percent Reduction
SIFT Files	197MB	
Reduced SIFT Files	4.88MB	97.5
SURF Files	290MB	
Reduced SURF Files	16.1MB	94.4

experiment is 24 hours 39 minutes on the unreduced SIFT keypoint files. The total time needed to complete the keypoint reduction and perform the matching experiment is 9 hours 50 minutes which is a reduction of 60.1%. The time required to match using

Table 4.4: Reduced and unreduced algorithm match execution time for SIFT.

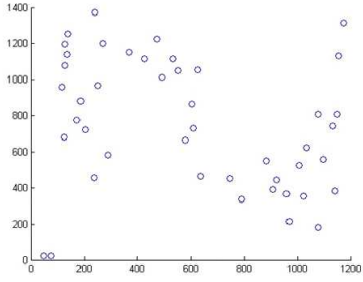
SIFT Algorithm	Approximate Execution Time	Percent Reduction
Match	24 hours 39 minutes	N/A
Reduced Match	6 hours 23 minutes	74.1
Keypoint Reduction	3 hours 27 minutes	86.0
Match and Keypoint Reduction	9 hours 50 minutes	60.1

the SURF algorithm is somewhat lower than the SIFT algorithm. Table 4.5 shows that the time needed to run the matching experiment on unreduced key files is 12 hours 19 minutes while the total time needed for the reduced key files is 3 hours 55 minutes. This is a 68.2% reduction in time. The system used to run the experiments is a dual core Xenon 3GHz workstation with 3GB of system RAM.

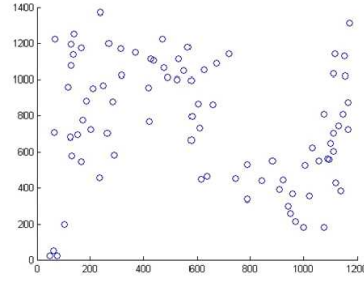
Table 4.5: Reduced and unreduced algorithm match execution time for SURF.

SURF Algorithm	Approximate Execution Time	Percent Reduction
Match	12 hours 19 minutes	N/A
Reduced Match	2 hours 16 minutes	81.6
Keypoint Reduction	1 hours 39 minutes	86.6
Match and Keypoint Reduction	3 hours 55 minutes	68.2

4.1.1 Testing Keypoints. Figure 4.3 is a sample of an image's keypoint distribution with 52 keypoints in Figure 4.3(a) and 102 keypoints in Figure 4.3(b). Both distributions were generated using a distance weight and a scale weight equal to 1, where the scale is based off the scale of the detected keypoint. As can be seen in this example, the larger the number of keypoints, the more uniform the distribution along both axes. Since the background is going to be occluded at times, the more uniform the distribution of points, the better the matching opportunities. Because of this, the number of keypoints is set to 102. A larger set of keypoints was considered, but the computational cost of the keypoint reduction is high so a decision was made



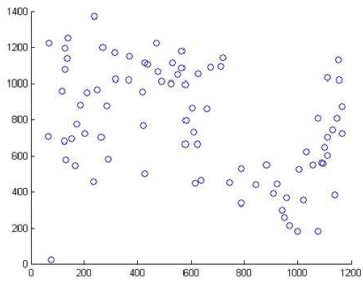
(a) Image with 52 keypoints



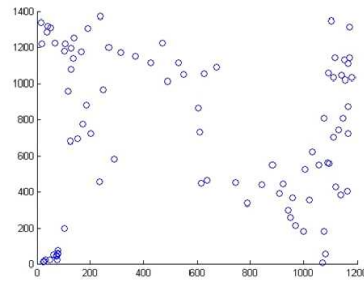
(b) Image with 102 keypoints

Figure 4.3: Example Keypoint Distribution. Axis are the x, y coordinate of pixels to limit the number of keypoints to 102. More research is required to find the optimal keypoint count.

Tests were conducted using just the value of the scale to determine the selected keypoints. The effect of this weighting can be seen in Figure 4.4 and in Figure 4.3(b). Figure 4.4(a) shows keypoint distribution with no distance calculation, thus selection is based on the scale value of the keypoints. Figure 4.3(b) shows keypoint distribution with distance weight equal 1, and Figure 4.4(b) shows the keypoint distribution with a distance weight equal to 5.



(a) distance:scale 0:1



(b) distance:scale 5:1

Figure 4.4: Two feature distributions of 102 keypoints with different weighting on distance. Axis are the x, y coordinate of pixels

4.2 SIFT Algorithm

Keypoint generation for SIFT uses a program developed by Rob Hess [8]. Windows operating system batch files are used to sequentially execute the keypoint generation program on the provided gray scale images. SIFT matches were computed for all 7750 possible unique compares plus 125 self matches for a total of 7875 compares.

Figure 4.5 shows that the SIFT match algorithm does very well with occlusion. For the two images matched in Figure 4.5, there are a total of 6 matches found. One of the matches, the one on the individual's arm, is an incorrect match, therefore SIFT had 5 good matches and one bad match. This is a trend for the SIFT algorithm. If we set a minimum threshold (η) of 5 *correct* matches then there is reasonable certainty that the background is a match. With the threshold (η) of 5 Figures 4.7 and 4.6 show there are a relatively small number of incorrectly matched images, with an accuracy of 81%, that are from different locations but are detected as being the same location. The lower resolution images matched poorly with an accuracy of 72.5%. With the current number of keypoints and the method of reducing the keypoints some matches were not detected. The SIFT algorithm has the highest accuracy at a threshold (η) of 6 with the maximum value of 81.1%. This algorithm correctly matches to the same picture even with a very large threshold (η) value of 98. The reason a threshold (η) value of 102 will incorrectly drop some identical image matches is the matching algorithm uses a nearest neighbor algorithm to find the keypoint matches and some of the neighbors are pruned during keypoint reduction [11]. Figure 4.6 shows that the Type I error (false positive) drops dramatically at a threshold (η) of 4. With the 125 reduced match set the number of correct matches is 16. The false positive rate is 1.15% and with the third party classification the rate changes slightly to 1.14%. The Type II (false negative) error rate is 57.9% and with the third party group partitioning again there is a small change 56.76%.

With the inclusion of the match comparison test which is the standard deviation, the accuracy of the reduced keypoint matches does not change significantly from the reduced keypoint matching accuracy. Figure 4.7 shows the SIFT accuracy with and



Figure 4.5: SIFT Image Showing The Reduced Keypoint Matches With Occlusion.

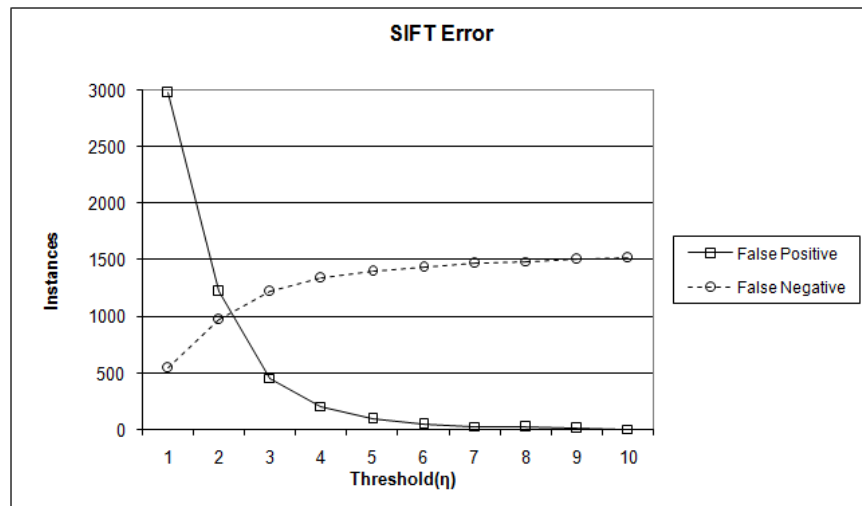


Figure 4.6: SIFT Error on Reduced Keypoints Showing the False Positives and False Negatives.

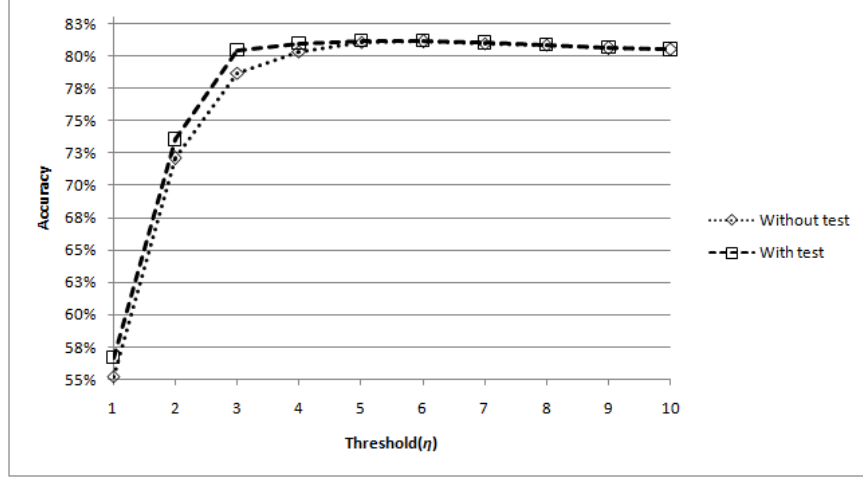


Figure 4.7: Reduced Keypoint SIFT Accuracy with and without Match Comparison Test.

without match comparison test. The reduced keypoint match accuracy is 81.12%, and the reduced keypoint match accuracy with match comparison test is 81.21%. There is an insignificant change in accuracy performance when the keypoints are reduced, as can be seen in Figure 4.1. The accuracy is 81.55% for unreduced keypoint matches and is 81.21% with the reduced keypoint match with match comparison test. The other match comparison test, where the intersections are tested on whether or not they occur within the match image frame, produced the same maximum accuracy of 81.21%. An example image Figure 4.8 depicting the match lines and showing the intersection points as small circles where the X indicates the average intersection position and the dotted line ellipse is the standard deviation distance.

4.3 SURF Algorithm

The SURF algorithm implementation by Herbert Bay, Andreas Ess, Geert Willems and the Windows port by Stefan Saur is used to produce the SURF key files [3]. Like the SIFT keypoint generation, batch files ensure the interest point and descriptor files are named properly and stored for later use. Like the SIFT algorithm SURF matching was computed using 7875 comparisons.

The SURF algorithm produces a larger number of matches. The smallest num-

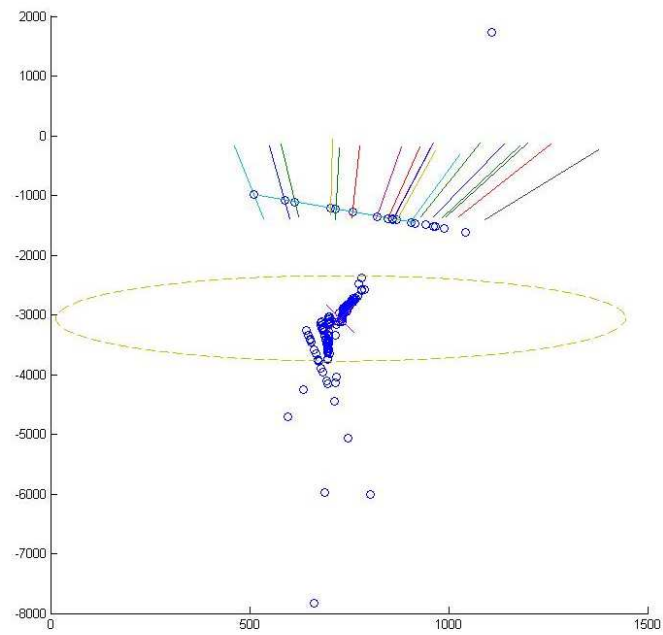


Figure 4.8: Match Comparison Test Showing Match Lines, Intersect Points (Circles), Average Intersect Point (X), And The Standard Deviation Distance (Dotted Line).

ber of matches is 16 and the algorithm produces an average 41 matches. Most of the matches range from 30 to 60. However, this algorithm produces a large percentage of mismatches. Figure 4.9 shows the SURF match image with several more mismatches than SIFT on the images in Figure 4.5. Figure 4.9 contains 44 total matches. Vi-

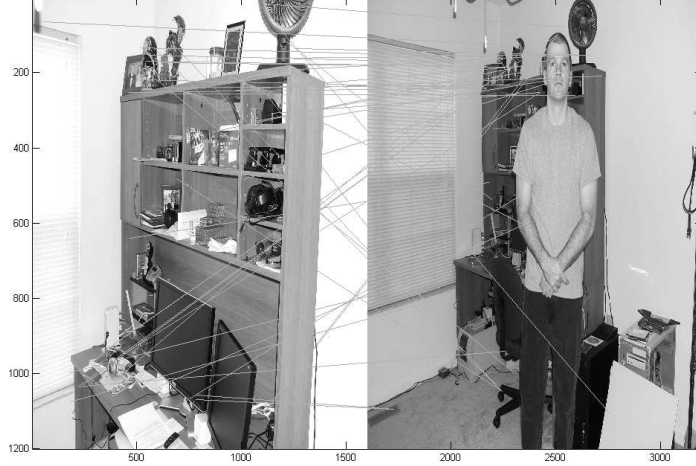


Figure 4.9: SURF Image Showing Reduced Keypoint Matches With Occlusion.

sual inspection makes it difficult to count the correct matches. Figure 4.10 shows the number of Type I and Type II errors are quite high. The SURF algorithm has

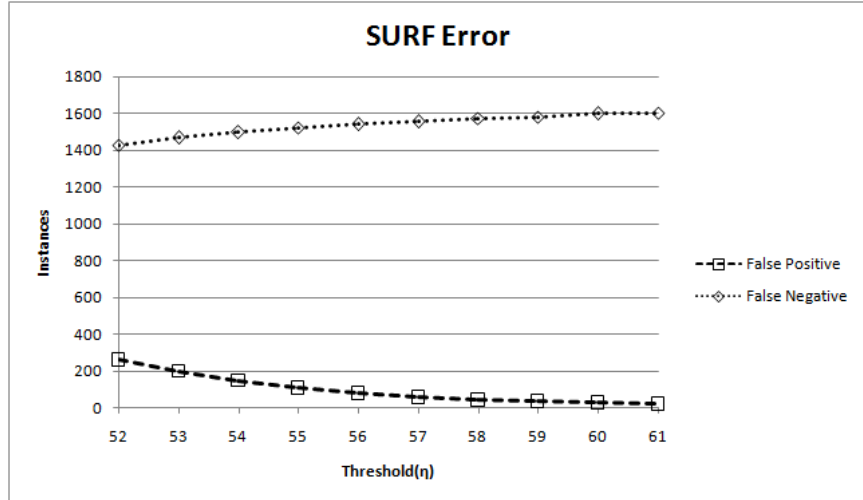


Figure 4.10: SURF Error on Reduced Keypoints Showing the False Positives and False Negatives.

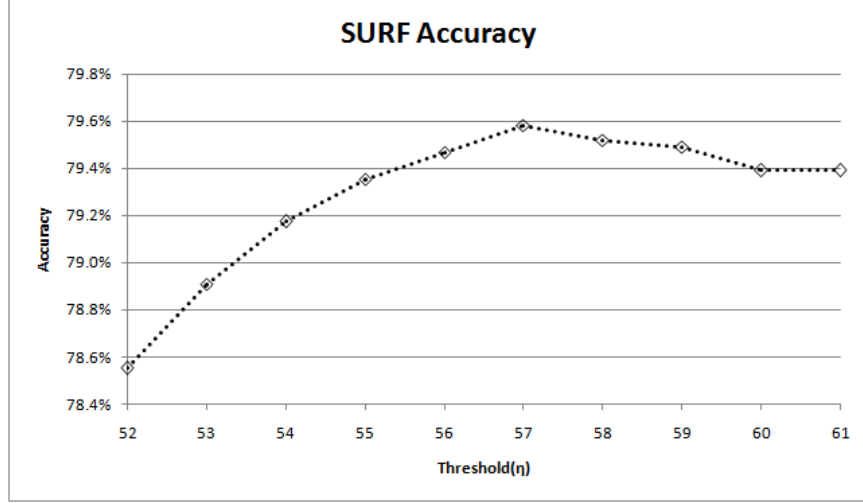


Figure 4.11: SURF Accuracy on Reduced Keypoints.

no problems matching against the same picture. Figure 4.11 shows the maximum accuracy of 79.58% occurs at $\eta = 57$ where the unreduced SURF accuracy is 78.26%.

The match comparison test does not significantly alter the accuracy of the SURF algorithm. Without the slope threshold added to the SURF match comparison test, the accuracy does not change from 79.58%. The standard deviation is so large no match lines are removed as bad. Accuracy, however, changes as a function of ϵ in (3.11). Two values were used, $\epsilon = 0.3$ and $\epsilon = 0.4$. These represent an angle of 16.7° and 21.8° respectively. The maximum accuracy of the SURF algorithm when $\epsilon = 0.4$ is 80.44% as shown in Figure 4.12. The maximum accuracy of the SURF algorithm when $\epsilon = 0.3$ is 80.75% shown in Figure 4.13. The maximum accuracy without the match comparison test with the SURF algorithm is 79.58%. With the match comparison test the maximum accuracy of SURF is 80.75%. These accuracies have not been shown to be statistically different.

4.4 Human Accuracy

For comparison, a third party individual classified the locations of the image test set. The individual had not seen any of the locations and was asked to place images into groups based on location. He was instructed to only place images into

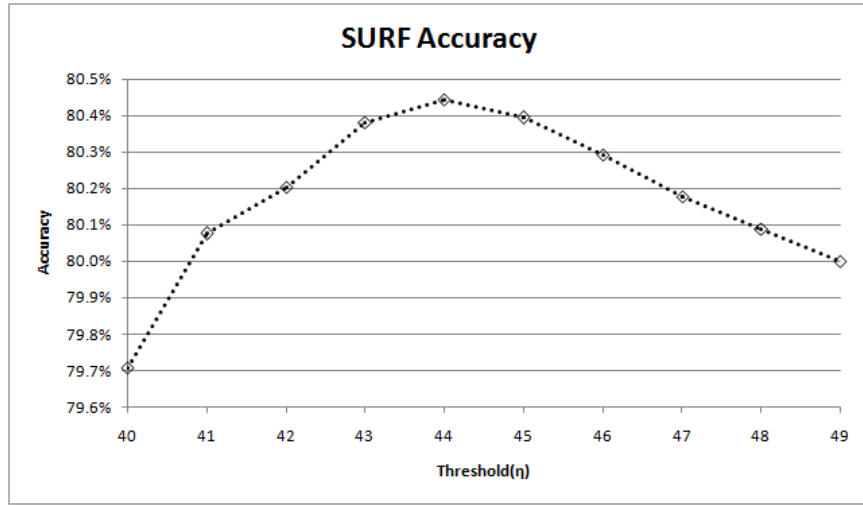


Figure 4.12: SURF Accuracy with $\epsilon = 0.4$.

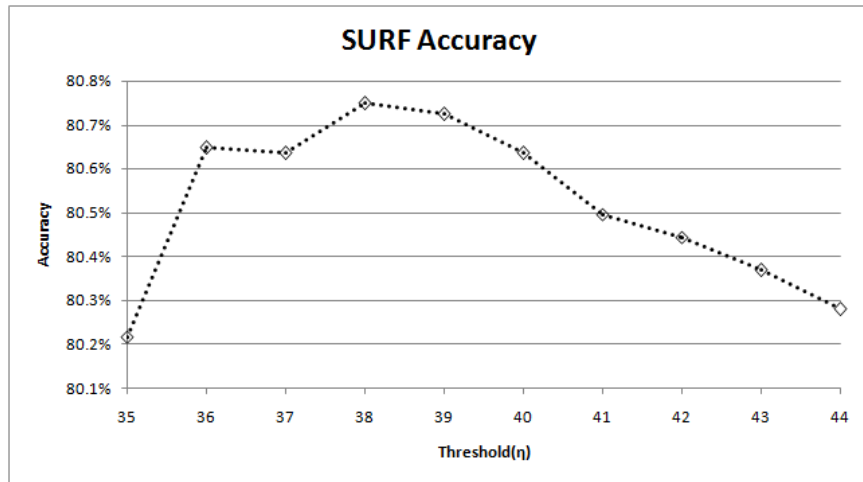


Figure 4.13: SURF Accuracy with $\epsilon = 0.3$.

the same group if there was no doubt that the image belonged in said group. The person divided the images into 24 different location groups using what he considered to be prominent reference points to distinguish image locations. Of the 24 groups, 6 contain a single image. Not being limited to the 7 actual image locations his accuracy was 55% due to him creating unnecessary groups. This demonstrate the difficulty of image data base matching for this data base. The individual did not perform the image matching the same way as the algorithms. He did not perform 7875 comparisons so direct comparison of the algorithm accuracies is not statistically relevant.

4.5 Shi-Tomasi Algorithm

This algorithm is implemented in the Kanade-Lucas-Tomasi Feature Tracker program or KLT developed by Stan Birchfield at Clemson University [5]. The KLT program was modified to allow command line input, and a batch file was used to produce the feature list files. Shi-Tomasi matching is reduced for a total of 125 compares. The first image was compared to all the other available images. The Shi-Tomasi algorithm performed poorly; it did not track a single feature to any of the images other than the first image matched with itself.

4.6 Results Conclusion

The maximum accuracy found for the two algorithms, SIFT and SURF shows no statistically significant change in accuracy after the keypoint reduction and match point comparison from the original unreduced keypoint matching. There was a marked reduction in size requirements and computation time when using the reduced keypoints with match point comparison for matching.

V. Conclusion

This research shows that automating the matching process of matching image backgrounds, to group them based on location, is indeed possible. Specifically, the SIFT algorithm, with our keypoint reduction technique using the Mahalanobis distance plus the scale of detected keypoint, has a similar accuracy to the other two techniques. The SIFT algorithm's maximum accuracy of 81.1% is attained using reduced keypoint files, but including the match comparison test the accuracy is 81.2%. SIFT's accuracy is 81.6% when using the unreduced keypoints. The space savings from using the reduced keypoint files is a 97.5% reduction for SIFT. The time savings when using the reduced keypoint files for SIFT is 60.1% and for SURF it is 68.2%. Using the reduced key file, SURF's maximum accuracy is 79.6% prior to the inclusion of the match comparison test. With the comparison test the accuracy is 80.8%. The accuracy of the SURF algorithm using the unreduced key files is 78.3%. The listed accuracies are not likely to be statistically significant. The Shi-Tomasi algorithm does not show any promise for this problem domain as it fails to match any locations other than the picture matching to itself. SIFT and SURF keypoint reduction do not degrade performance and provides a speedup in processing. With more research, both SIFT and SURF appear to be useful within the research problem domain.

5.1 Future Work

The research shows that more work can be done to further automate the process. Work needs to be completed on analyzing the match points from the matching images to improve the accuracy. Specifically, the optimal value of ϵ , used in the match comparison test for SURF, needs to be found. The percentage threshold value of intersect points that fall outside the standard deviation also need to be optimized. Testing needs to be run on different realistic image data bases of varying content and quality. The number of keypoints selected in the keypoint reduction step needs to be optimized with respect to the data base size, matching speed, and matching

accuracy. Also, other feature generation and matching algorithms, such as those in [15] [21] can be used in both the SIFT and SURF algorithm's match comparison test. A technique for rejecting incorrect matches rejects multiple matches that come from the same keypoint. It is known a keypoint should only match with one other keypoint in an image. Any point that produces multiple matches should be pruned. Additional image data bases are also needed. These image data bases need to have the image properties stored with the images. For example, the 3 dimensional location of the camera in the room, the pitch, rotation, and direction of the camera, need to be recorded at the time the images are taken. Also more data needs to be collected; specifically, the number of keypoints that are retained should be varied to test whether matching accuracy versus storage size can be improved. The importance of the match time should also be taken into account. This may also assist with the matching of lower resolution images to higher resolution images.

Bibliography

1. Alhichri, Haikel Salem and Mohamed Kamel. “Virtual circles: a new set of features for fast image registration”. *Pattern Recogn. Lett.*, 24(9-10):1181–1190, 2003. ISSN 0167-8655.
2. Ashley, Wayne. “What shoe was that? The use of computerised image database to assist in identification”. *Forensic Science International*, 82(1):7–20, 9/15 1996.
3. Bay, Herbert, Luc Van Gool, and Tinne Tuytelaars. “SURF: Speeded Up Robust Features software”. <http://www.vision.ee.ethz.ch/surf/index.html>.
4. Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. “SURF: Speeded Up Robust Features”. *Proceedings of the ninth European Conference on Computer Vision*. May 2006.
5. Birchfield, Stan. “KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker”. <http://www.ces.clemson.edu/stb/klt/>.
6. Gonzalez-Rodriguez, J., J. Fierrez-Aguilar, D. Ramos-Castro, and J. Ortega-Garcia. “Bayesian analysis of fingerprint, face and signature evidences with automatic biometric systems”. *Forensic science international*, 155(2-3):126–140, 2005.
7. Haibin Ling; Jacobs, D.W. “Deformation invariant image matching”. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2:1466–1473 Vol. 2, 17-21 Oct. 2005. ISSN 1550-5499.
8. Hess, Rob. “SIFT software”. <http://web.engr.oregonstate.edu/hess/>.
9. KUTULAKOS, K. “SIFT Lecture Notes.” <http://www.cs.toronto.edu/kyros/courses/320/Lectures.s07/lecture.2007s.24.pdf>, 2007.
10. Lowe, David G. “Object Recognition from Local Scale-Invariant Features”. *ICCV ’99: Proceedings of the International Conference on Computer Vision-Volume 2*, 1150. IEEE Computer Society, Washington, DC, USA, 1999. ISBN 0-7695-0164-8.
11. Lowe, David G. “Distinctive Image Features from Scale-Invariant Keypoints”. *Int. J. Comput. Vision*, 60(2):91–110, 2004. ISSN 0920-5691.
12. Matas, Jiri, Stepan Obdrzalek, and Ondrej Chum. “Local Affine Frames for Wide-Baseline Stereo”. *icpr*, 04:40363, 2002. ISSN 1051-4651.
13. Murtagh, Zeno J. Geradts; Jurrien Bijhold; Rob Hermesen; Fionn. “Image matching algorithms for breech face marks and firing pins in a database of spent cartridge cases of firearms”. *Forensic science international*, 2001., 119(1):97–106, June 2001.
14. Noble, J. Alison. “Finding corners”. *Image Vision Comput.*, 6(2):121–128, 1988. ISSN 0262-8856.

15. Sebe, N. and M. Lew. “Comparing Salient Points Detectors”, 2003. URL citeseer.ist.psu.edu/sebe01comparing.html.
16. Shi, Jianbo and Carlo Tomasi. “Good features to track”. *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994.Proceedings CVPR '94.*, 593, -06-23 1994. Doi: pmid:.
17. Stefan Zickler, Alexei Efros. “Detection of Multiple Deformable Objects using PCA-SIFT”. 2007.
18. Veth, M. and J. Raquet. “Fusion of Low-Cost Imaging and Inertial Sensors for Navigation”. *Proceedings of the GNSS 2006 Conference*. Institute of Navigation, Fort Worth, TX, September 26-29 2006.
19. Veth, M. and J. Raquet. “Two-Dimensional Stochastic Projections for Tight Integration of Optical and Inertial Sensors for Navigation”. *Proceedings of the National Technical Meeting*. Institute of Navigation, Monterey, CA, January 18-20 2006.
20. Williams, Luke Ledwich; Stefan. “Reduced SIFT Features For Image Retrieval and Indoor Localisation”. *Australian Conference on Robotics and Automation*. Australian Robotics and Automation Association, Canberra, Australia, December 2004.
21. Winder, Matthew, Simon A. J.; Brown. “Learning Local Image Descriptors”. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 1–8, 17-22 June 2007.
22. Zhao, W. “Face recognition: A literature survey”. *ACM Computing Surveys (CSUR)*, 35(4), -12 2003. Doi: pmid:.
23. Zitova, Barbara and Jan Flusser. “Image registration methods: a survey”. *Image and Vision Computing*, 21(11):977–1000, October 2003. URL [http://dx.doi.org/10.1016/S0262-8856\(03\)00137-9](http://dx.doi.org/10.1016/S0262-8856(03)00137-9).

REPORT DOCUMENTATION PAGE					<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.						
1. REPORT DATE (DD-MM-YYYY) 20-12-2007		2. REPORT TYPE Master's Thesis			3. DATES COVERED (From — To) May 2006 — Dec 2007	
4. TITLE AND SUBTITLE <div style="text-align: center;">Forensics Image Background Matching Using Scale Invariant Feature transform (SIFT) And Speeded Up Robust Features (SURF)</div>					5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER 5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER 	
6. AUTHOR(S) Paul N. Fogg II					8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GCE/ENG/08-02	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management 2950 Hobson Way WPAFB OH 45433-7765					10. SPONSOR/MONITOR'S ACRONYM(S) DC3/DCCI	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) DoD Cyber Crime Center/Defense Cyber Crime Institute Mr. Ed Kung 911 Elkridge Landing Rd. Linthicum, MD 21090 Commercial (410)981-1169					11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approval for public release; distribution is unlimited.						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT In criminal investigations, it is not uncommon for investigators to obtain a photograph or image that shows a crime being committed. Additionally, thousands of pictures may exist of a location, taken from the same or varying viewpoints. Some of these images may even include a criminal suspect or witness. One mechanism to identify criminals and witnesses is to group the images found on computers, cell phones, cameras, and other electronic devices into sets representing the same location. One or more images in the group may then prove the suspect was at the crime scene before, during, and/or after a crime. This research extends three image feature generation techniques, the Scale Invariant Feature Transform (SIFT), the Speeded Up Robust Features (SURF), and the Shi-Tomasi algorithm, to group images based on location. The image matching identifies keypoints in images with changes in the contents, viewpoint, and individuals present at each location. After calculating keypoints for each image, the algorithm stores the strongest features for each image are stored to minimize the space and matching requirements.						
15. SUBJECT TERMS image matching, SIFT, SURF, Shi-Tomasi						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT		18. NUMBER OF PAGES	
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU		60	
U	U	U				
19a. NAME OF RESPONSIBLE PERSON Dr. Gilbert L. Peterson					19b. TELEPHONE NUMBER (include area code) (937)785-6565ext4281, Gilbert.Peterson@afit.edu	